
Übung 2

Abgabe bis **Donnerstag, 14. April 10:00** via EPIIC: <http://ep.iic.jku.at>.

1. MIPS Assembler (12 Punkte)

Die *Ackermann Funktion* is rekursiv definiert durch:

$$A(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ A(m - 1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

Übersetze die Definition zuerst in eine rekursive C Funktion, und diese danach in MIPS. Berücksichtige dabei die Konventionen bezüglich Registerbelegung. Führe die MIPS Funktion mit den Parametern $m = 3$ und $n = 4$ aus. Die Abgabe beinhaltet das kommentierte MIPS Programm und einen Screenshot von der Ausführung in *QtSpim*.

2. Maschinensprache (6 Punkte)

Es soll ein neuer Rechner entworfen werden. Die Rechengenauigkeit aller Operation soll 12 Bit betragen. Der Befehlssatz soll 20 Instruktionen beinhalten. Der Rechner hat 14 Register bei einem Adressraum von maximal 18K. Es wird eine Load/Store Architektur verwendet: Befehle, die auf den Speicher zugreifen haben zwei Register als Operanden. Kein Befehl hat mehr als drei Operanden, mindestens zwei Operanden sind Register und der dritte kann eine 8-Bit-Konstante oder ein Register sein.

- (a) Wie breit müssen die Register mindestens sein?
- (b) Wie viele Bit werden benötigt, um ein Maschinenwort zu codieren?

Begründe deine Überlegungen.

3. Befehlskodierung (6 Punkte)

Das folgende Programm soll auf dem Rechner aus Aufgabe 2 ausgeführt werden.

```
add  $A, $B, $A    # A = A+B
subi $C, $X, 9     # C = X-9
load $C, $A        # C = Mem[A]
add  $B, $D, $C    # B = D+C
addi $X, $X, 4     # X = X+4
subi $C, $B, 5     # C = B-5
```

Gib eine mögliche Binäre Kodierung der Befehle des Assembler Programms an und beschreibe die getroffenen Annahmen.